



Adaptivity in Bayesian inverse finite element problems

Learning and simultaneous control of discretisation and sampling errors

Kerfriden, Pierre; Kundu, Abhishek; Claus, Susanne

Published in:
Materials

DOI:
[10.3390/ma12040642](https://doi.org/10.3390/ma12040642)

Publication date:
2019

Document version
Publisher's PDF, also known as Version of record

Document license:
[CC BY](#)

Citation for published version (APA):
Kerfriden, P., Kundu, A., & Claus, S. (2019). Adaptivity in Bayesian inverse finite element problems: Learning and simultaneous control of discretisation and sampling errors. *Materials*, 12(4), [642].
<https://doi.org/10.3390/ma12040642>

Article

Adaptivity in Bayesian Inverse Finite Element Problems: Learning and Simultaneous Control of Discretisation and Sampling Errors

Pierre Kerfriden ^{1,2,*} , Abhishek Kundu ¹  and Susanne Claus ^{1,3}

¹ School of Engineering, Cardiff University, The Parade, Cardiff CF243AA, UK; KunduA2@cardiff.ac.uk (A.K.); ClausS2@cardiff.ac.uk (S.C.)

² MINES ParisTech, PSL University, Centre des Matériaux, BP87 91003 Evry, France

³ Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark

* Correspondence: KerfridenP@cardiff.ac.uk or Pierre.Kerfriden@mines-paristech.fr

Received: 23 December 2018; Accepted: 5 February 2019; Published: 20 February 2019



Abstract: The local size of computational grids used in partial differential equation (PDE)-based probabilistic inverse problems can have a tremendous impact on the numerical results. As a consequence, numerical model identification procedures used in structural or material engineering may yield erroneous, mesh-dependent result. In this work, we attempt to connect the field of adaptive methods for deterministic and forward probabilistic finite-element (FE) simulations and the field of FE-based Bayesian inference. In particular, our target setting is that of exact inference, whereby complex posterior distributions are to be sampled using advanced Markov Chain Monte Carlo (MCMC) algorithms. Our proposal is for the mesh refinement to be performed in a goal-oriented manner. We assume that we are interested in a finite subset of quantities of interest (QoI) such as a combination of latent uncertain parameters and/or quantities to be drawn from the posterior predictive distribution. Next, we evaluate the quality of an approximate inversion with respect to these quantities. This is done by running two chains in parallel: (i) the approximate chain and (ii) an enhanced chain whereby the approximate likelihood function is corrected using an efficient deterministic error estimate of the error introduced by the spatial discretisation of the PDE of interest. One particularly interesting feature of the proposed approach is that no user-defined tolerance is required for the quality of the QoIs, as opposed to the deterministic error estimation setting. This is because our trust in the model, and therefore a good measure for our requirement in terms of accuracy, is fully encoded in the prior. We merely need to ensure that the finite element approximation does not impact the posterior distributions of QoIs by a prohibitively large amount. We will also propose a technique to control the error introduced by the MCMC sampler, and demonstrate the validity of the combined mesh and algorithmic quality control strategy.

Keywords: finite element inverse problems; Bayesian statistics; Data-driven modelling; error estimation; MCMC; machine learning

1. Introduction

The Bayesian statistical framework has been used extensively in the problem of system identification [1] or model updating based on experimental test data [2,3]. The objective of using inverse problems to learn or calibrate model or latent parameters, including model error terms, lies in the fact that underlying parametrised computational models are uncertain, or erroneous [4]. The experimental data when assimilated into the model, using the Bayesian inference framework, is expected to provide joint estimates of the model parameters conditional on the data and compensate for uncertainty/bias

in the model predictions. However, operating only in the field of model parameters is problematic for a number of reasons. Firstly, the underlying computational models for any practical application is expensive. As a result working with a very high resolution numerical model at every stage is prohibitively expensive. Secondly, when using surrogate models in the parameter space in conjunction with a simulator, adaptive enrichment of the response surface does not take adaptive mesh refinement within its purview. This is not optimal since ensuring that the model error in the energy norm is bounded in the parameter space requires a uniformly high resolution and increases the associated computational overhead. Lastly the advanced adaptive mesh refinement techniques are rarely used in conjunction with parametric learning using Bayesian inference. There is significant room for improvement in this regard since a simultaneous control of both statistical and discretisation error would lead to substantially improved predictive numerical models both in terms of accuracy and computational efficiency.

At the root of our methodology is the estimation of errors due to the finite element approximation of the partial differential equation (PDE) of interest. Classically, spatial resolution of finite element models can be adaptively refined (also known as local h -refinement) based on a posteriori error estimation techniques [5–7], combined with re-meshing strategies. Within this, methods focussing on errors estimated in terms of specific quantities of interest (QoI), rather than the classical energy norm, constitute the goal-oriented adaptivity scheme [8–12], which is of particular interest in the present study. Numerical studies have shown that these give better convergence in the local features of the solution compared to traditional approaches.

Integrating model reduction techniques with finite element model updating techniques has received some attention in recent years [13], where the motivation is to use Bayesian model updating framework with an adaptive scheme of enriching the surrogate response surface. Multistage Bayesian inverse problems are quite important in this respect [14,15] and have important applications for system identification of vibrating systems. The prediction error is an important parameter to be calibrated in such cases, as the authors point out. But the improvement in model predictions, if solely focussed on obtaining the posterior probabilistic parameter estimates or for adaptive enrichment of the response surface in the parameter space, without considering simultaneous enhancement of the resolution of the numerical simulator would be unsatisfactory both from the standpoints of computational accuracy and efficiency.

The forward problem of uncertainty propagation has been investigated extensively for the solution of stochastically parametrised partial differential equations. These range from efficient stochastic Galerkin methods using polynomial chaos basis functions [16–19], stochastic collocation techniques [20,21], Monte-Carlo sampling based methods (and its various improvements) [22–24] and other deterministic sampling methods [25–27]. The main challenge is to obtain a good approximation of the lower order statistical moments of the state vector or specific quantities of interest. On the other hand, advances in the resolution of stochastic inverse problems has become a very active topic in engineering and mathematical research (see e.g., [28]). Scalable Bayesian inversion algorithms for large-scale problems have been investigated [29] and as well as Bayesian inversion to probabilistic robust optimization under uncertainty [30]. The use of adaptive sparse-grid surrogates unified with Bayesian inversion for posterior density estimates of the model or design has also been studied parameters [3,31,32]. This research focuses mostly on the definition of surrogates of the numerical model and having adaptive methods to control the statistical sampling error in the definition of the response surface. Lately there has been some research [33] that defines adaptivity as the local enrichment of the surrogate model and uses error estimation to bound this error, with no mention of spatial discretisation. However, coupling engineering uncertainty quantification (UQ) with an adaptive scheme for goal-oriented finite element model refinement remains a sparsely studied domain and presents significant challenges. It is so both from the problem formulation perspective, owing to the choice of appropriate candidate estimates based on which adaptive model enrichment can be performed, as well as incorporating it into the general formulation of Bayesian inversion.

The main focus of the paper is to develop a robust methodology for the simultaneous control of errors from multiple sources—the goal-oriented finite element error and the uncertainty-driven statistical error—in a Bayesian identification framework for identification system parameters conditional on data (experimental or otherwise). The novel algorithmic approach proposed in this paper consists in running two Markov Chain Monte Carlo algorithms simultaneously to sample the posterior densities of the quantities of interest (component-wise MCMC). The first chain utilises the current finite element model, whilst the second chain runs with a corrected likelihood function that takes into account the discretisation error. The latter quantity may be constructed by making use of a posteriori finite element error estimates available in the literature. At any time during the sampling process, two empirical densities are available and may be compared to evaluate the effect of the discretisation error onto posterior densities. The second building block of our algorithmic methodology allows us to determine whether enough samples have been drawn by the component-wise MCMC. Using multiple parallel chains combined with bootstrap-based estimates of sampling errors, we automatically stop the MCMC algorithms when either (i) the required level of accuracy is achieved or (ii) we have generated enough statistical confidence in the fact that the discretisation error is too large for our purpose, and therefore that mesh refinement is necessary. While any available error estimate of the discretisation error may be used within the general strategy outlined above (a goal-oriented residual or recovery-based error estimate, for instance [8,9,11,34]), we choose instead to construct this estimate via a dedicated machine learning approach. This interesting feature, inspired by previous work in data-driven error modelling [35–38], will be briefly outlined in the paper.

The paper is organised as follows. Section 2 introduces the Bayesian inverse problem based on a finite element model of a parametrised vibrating structure. This section also includes discussions on discretisation error. Some numerical examples are presented in Section 3, which aims to demonstrate the convergence of the joint posterior distributions on the model parameters through successive stages of mesh refinement. Section 4 discusses the total error as a combination of statistical and discretisation error that results from the MCMC algorithm used for sampling from posterior distributions. Section 5 gives the methodology for robust, simultaneous control of all error sources within the adaptive inverse problem solver using a component-wise MCMC algorithm in conjunction with bootstrap-aggregated regression model for model parameters. Numerical examples are presented and discussed in Section 6 to demonstrate the capabilities of the proposed methodology.

2. Finite Element Bayesian Inverse Problems

Although the methodology proposed in this paper is general, we will apply it to the verification of popular inverse finite element procedures used to monitor the integrity of structures during service. We will assume that the structure can be modelled as an elastic body, and that potential structural damage can be modelled by the evolution of a field of elastic constants characterised by a finite number of uncertain parameters. Assuming that the resonance frequencies of the structure have been measured physically, we attempt to identify this set of mathematical parameters, through the solution of an inverse problem. Significant deviations from the parameter values corresponding to an undamaged structural state may reveal structural failure. This is a typical condition monitoring method used for non-destructive structural testing.

In order to clarify the proposed study, we will assume that we have measured the n_d first dynamic eigenfrequencies $d = (\ln(\omega_1) \quad \dots \quad \ln(\omega_{n_d}))^T$ of the structure represented in Figure 1. The purpose of the model inversion is to identify the values of n_μ parameters $\mu = (\mu_1 \quad \dots \quad \mu_{n_\mu})^T$ of the structural dynamics model, here the log of the elastic moduli of the subdomains represented in Figure 1. We will also aim to predict the remaining $n_p - n_d$ frequencies $p = (\ln(\omega_{n_d+1}) \quad \dots \quad \ln(\omega_{n_d+n_p}))^T$. The relationships $\mu \rightarrow d(\mu)$ and $\mu \rightarrow p(\mu)$, i.e., the computational model, are defined

implicitly through the evaluation of a standard finite element model of the steady-state, undamped structural vibrations.

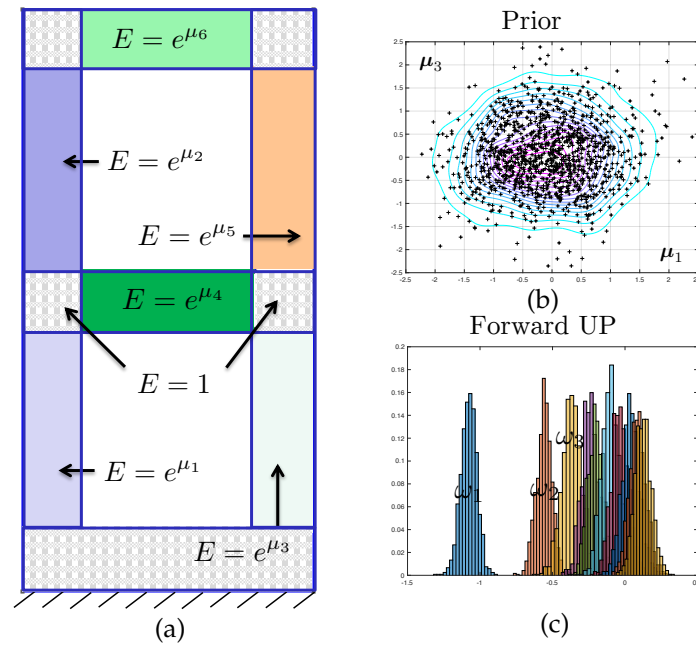


Figure 1. (a) Structure undergoing structural vibrations. The grey area have a known, deterministic Young modulus. This property is piecewise constant in the coloured areas, and Gaussian distributed. (b) The corresponding marginal probability density function for two of the 6 parameters is represented in the top-left corner. (c) Standard Monte-Carlo finite element process is employed to compute the distribution of the 10 first free vibration frequencies and represent the marginal distribution of each of these quantities.

2.1. Bayesian Inverse Problem

We assume that the quantities d that are measured experimentally are described by the mathematical model, up to an error, which is modelled in a probabilistic manner. Following standard Bayesian procedures, this error is modelled as a white Gaussian noise,

$$d = d_m + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \Sigma_n^{-1}) \quad (1)$$

with Σ_0 a positive definite covariance matrix. $A_d : \mu \rightarrow d_m := A_d(\mu)$ is the mathematical prediction that corresponds to the experimental observation. Model A_d is a deterministic function of n_μ uncertain model parameters that we organise in a n_μ -dimensional vector μ . Bayesian inversion requires to associate a prior probability density with the model parameters μ . This prior probability density, denoted by $\pi_\mu(\mu)$ in the following, encodes the knowledge that we possess about μ before making any physical observations.

Seen from a different point of view, the probabilistic inversion setting amounts to the definition of a joint probability distribution for μ and d :

$$\pi_{\mu,d}(\mu, d) = \pi_{d|\mu}(\mu, d) \pi_\mu(\mu) \quad (2)$$

The formal expression of likelihood function $\mathcal{L}(\mu; d) := \pi_{d|\mu}(\mu, d)$ is a direct consequence of assumption (1), namely

$$\mathcal{L}(\mu; d) = \frac{1}{\sqrt{(2\pi)^{n_d} |\Sigma_n|}} e^{-\frac{1}{2} \left((d - A_d(\mu))^T \Sigma_n^{-1} (d - A_d(\mu)) \right)} \quad (3)$$

It is now possible to formally proceed to the inversion itself by conditioning the joint distribution to actually observed quantities. Applying Bayes' formula, the posterior probability density of the model parameters is

$$\pi_{\mu|d}(\mu; d) = \frac{\mathcal{L}(\mu; d) \pi_{\mu}(\mu)}{\pi_d(d)} \quad (4)$$

where $\pi_d(d)$ is a normalising constant, whose computation requires a usually intractable integration. Bayes formula provides us with an updated knowledge about the uncertain part of our mathematical model. It is now possible to predict unobserved quantities i.e.,

$$p = A_p(\mu), \quad \mu \sim \pi_{\mu|d}(\mu; d) \quad (5)$$

by formally propagating the posterior uncertainty through model $A_p : \mu \rightarrow p_m := A_p(\mu)$. The posterior predictive probability density of p will be denoted by symbol $\pi_{p|d}(\mu; d)$.

2.2. Finite Element Modelling of Inverse Structural Vibration Problems

2.2.1. Direct Finite element Procedure for Frequency-Domain Vibrations

The numerical model

$$\mu \rightarrow A(\mu) := \begin{pmatrix} A_d(\mu) \\ A_p(\mu) \end{pmatrix} \quad (6)$$

is implicitly defined through the solution of a continuum mechanics problem. The components of d_m and p are the logarithm of the eigenvalues corresponding to the following parametrised eigenvalue problem: Find $u \in \mathcal{H}^1(\Omega)$ and $\omega \in \mathbb{R}^+$ such that $\forall v \in \mathcal{H}^1(\Omega)$

$$\int_{\Omega} \nabla_s u : \mathbb{D}(x; \mu) : \nabla_s v \, d\Omega - \omega^2 \int_{\Omega} \rho u \cdot v \, d\Omega = 0 \quad (7)$$

In the previous variational statement, Ω is the domain occupied by the structure of interest, $\mathcal{H}^1(\Omega)$ is the space of functions defined over Ω , with values in \mathbb{R}^2 , that are zero on part $\partial_u \Omega$ of the boundary of the domain, and whose derivatives up to order one are square integrable. ∇_s denotes the symmetric part of the gradient operator. \mathbb{D} is the fourth order Hooke tensor and ρ is the mass density of the solid material. The problem possesses an infinite number of solutions (ω, u) called free vibration modes. We order the free vibration frequencies in increasing order $\omega_1 < \omega_2 < \dots < \omega_{n_d+n_p}$. The free vibration modes are functions of uncertain material parameters μ through the definition of the Hooke tensor. Specifically,

$$\mathbb{D}(x; \mu) : \nabla_s \cdot = \lambda(x; \mu) \text{Tr}(\nabla_s \cdot) \mathbb{I} + 2 G(x; \mu) (\nabla_s \cdot) \quad (8)$$

with the parametrised Lamé constants

$$\lambda(x; \mu) = \frac{0.3 \times E(x; \mu)}{2(1 + 0.3)(1 - 2 \times 0.3)} \quad \text{and} \quad G(x; \mu) = \frac{E(x; \mu)}{2(1 + 0.3)} \quad (9)$$

Solving the continuum mechanics model is equivalent to evaluating the mapping

$$\mu \rightarrow \begin{pmatrix} A_d(\mu) \\ A_p(\mu) \end{pmatrix} = \begin{pmatrix} \ln(\omega_1(\mu) \quad \dots \quad \omega_{n_d}(\mu))^T \\ \ln(\omega_{n_d+1}(\mu) \quad \dots \quad \omega_{n_d+n_p}(\mu))^T \end{pmatrix}. \quad (10)$$

There is, in general, no analytical solution to the continuous vibration problem, and a standard way to obtain approximate solutions is to substitute a finite element space $\mathcal{U}^h(\Omega)$ for infinite dimensional search space $\mathcal{H}^1(\Omega) \supset \mathcal{U}^h(\Omega)$ [39]. Too coarse a finite element discretisation may result in poorly

predictive results, while too fine a mesh will lead to numerically intractable results, or, in any case, to a waste of computing resources.

2.2.2. Finite Element Approximation of the Bayesian Inverse Problem

Whilst using the continuum mechanics model exactly would deliver the posterior density $\pi_{\mu|d}(\mu; d)$ as solution of the Bayesian inverse problem, we now have an approximate posterior density

$$\bar{\pi}_{\mu|d}(\mu; d) = \frac{\bar{\mathcal{L}}(\mu; d) \pi_{\mu}(\mu)}{\bar{\pi}(d)} \quad (11)$$

where the finite element likelihood $\bar{\mathcal{L}}$ is obtained by substituting finite element mapping \bar{A}_d for A_d in Equation (3). Similarly, the approximate posterior density of p is denoted by $\bar{\pi}_{p|d}(\mu; d)$ and obtained by evaluating finite element mapping \bar{A}_p instead of A_p when propagating the posterior uncertainties forward.

2.2.3. Discretisation Error

The finite element error is the mismatch between $\bar{\pi}_{\mu|d}(\mu; d)$ and $\pi_{\mu|d}(\mu; d)$ on the one hand, and $\bar{\pi}_{p|d}(\mu; d)$ and $\pi_{p|d}(\mu; d)$ on the other hand. Various measures can be used to quantify this mismatch, amongst which the Hellinger distance, defined by

$$\mathcal{D}_h(\pi_{\mu|d}(\mu), \bar{\pi}_{\mu|d}(\mu)) = \left\| \sqrt{\pi_{\mu|d}(\mu)} - \sqrt{\bar{\pi}_{\mu|d}(\mu)} \right\|_2 \quad (12)$$

the Kullback-Leibler divergence, the total variation distance and the Kolmogorov-Smirnov (KS) distance, defined by

$$\mathcal{D}_{ks}(\pi_{\mu|d}(\mu), \bar{\pi}_{\mu|d}(\mu)) = \sup_{\mu} \left(\left| (\Pi_{\mu|d}(\mu) - \bar{\Pi}_{\mu|d}(\mu)) \right| \right) \quad (13)$$

where the capital symbols Π denotes the cumulative probability density corresponding to π , and $\bar{\Pi}$ is the finite element approximation of Π . This contribution will make use of the latter measure, in a one-dimensional setting (i.e., it will be applied to control the accuracy of the posterior density of one of the elements of μ or one of the elements of p). The attractiveness of the Kolmogorov–Smirnov distance is its straightforward application in the context of Monte-Carlo procedures, where only empirical densities are available, and its closeness to confidence intervals (CIs), which makes its values relatively easy to interpret within the context of a posteriori error estimation. Notice that the \mathcal{D}_{ks} is lower bounded by 0 (identical density functions) and upper bounded by 1 (non-overlapping support for the probability density functions).

3. Numerical Examples—Part I: Effect of Discretisation Errors Onto Posterior Densities

This section introduces the numerical examples that will be investigated in this paper, and aims to provide a first qualitative understanding of the effect of mesh refinement onto the quality of posterior probability densities.

3.1. Forward Stochastic Model

The stochastic field of Young's modulus that will be used to exemplify the error control approach proposed in this paper is defined via a decomposition of domain Ω into $n_{\mu} + 1 = 7$ non-overlapping subdomains $\{\Omega_i\}_{i \in [1, n_{\mu}+1]}$ such that $\bigcup_{i \in [1, n_{\mu}+1]} \Omega_i = \Omega$ and $\Omega_i \cap \Omega_j = \{\}$ for $i \neq j$. The domain decomposition is represented in Figure 1. Then, the proposed model is such that

$$E(x; \mu) = e^{\mu_i} \quad \text{if } x \in \Omega_i \text{ and } i \neq 1, \quad E(x; \mu) = 1 \quad \text{otherwise.} \quad (14)$$

Hence, the scalar parameters contained in vector $\mu \in \mathbb{R}^{n_\mu}$ are the logarithms of the Young's moduli corresponding to each of the subdomains.

The prior density is a multivariate Gaussian, and is given by equation

$$\pi_\mu(\mu) = \frac{1}{\sqrt{(2\pi)^{n_\mu} |\Sigma_0|}} e^{-\frac{1}{2} \left((\mu - \mu_0)^T \Sigma_0^{-1} (\mu - \mu_0) \right)} \quad (15)$$

The prior mean μ_0 is the null vector, and the prior variance $\Sigma_0 = \sigma_0^2 I_d$, where $I_d \in \mathbb{R}^{n_\mu} \times \mathbb{R}^{n_\mu}$ is the identity matrix, is diagonal and isotropic. The prior density is represented in dimensions (μ_1, μ_3) in Figure 1.

Finally, we model the error ϵ as a zero-mean multivariate Gaussian (consistently with what was described in previous section), with independent components and isotropic variance, i.e., $\Sigma_n = \sigma_n^2 I_d$.

3.2. Computational Meshes

The evaluation of the likelihood function appearing in solution (11) of the Bayesian inverse problem requires solving the continuum mechanics problem using the finite element method. In this example, we use a sequence of meshes $\{\mathcal{M}_i\}_{i \in \llbracket n_m \rrbracket}$ associated with a monotonically increasing number of degrees of freedom. These meshes are represented in Figure 2. Although the sequence of meshes is not strictly hierarchical, we see that the typically uniform element size is divided by ≈ 2 when moving from mesh \mathcal{M}_i to mesh \mathcal{M}_{i+1} . The intermediate meshes $\{\mathcal{M}_{i+\frac{1}{2}}\}_{i \in \llbracket 1 \tilde{n}_m \rrbracket}$ represented in Figure 2 will be used later on.

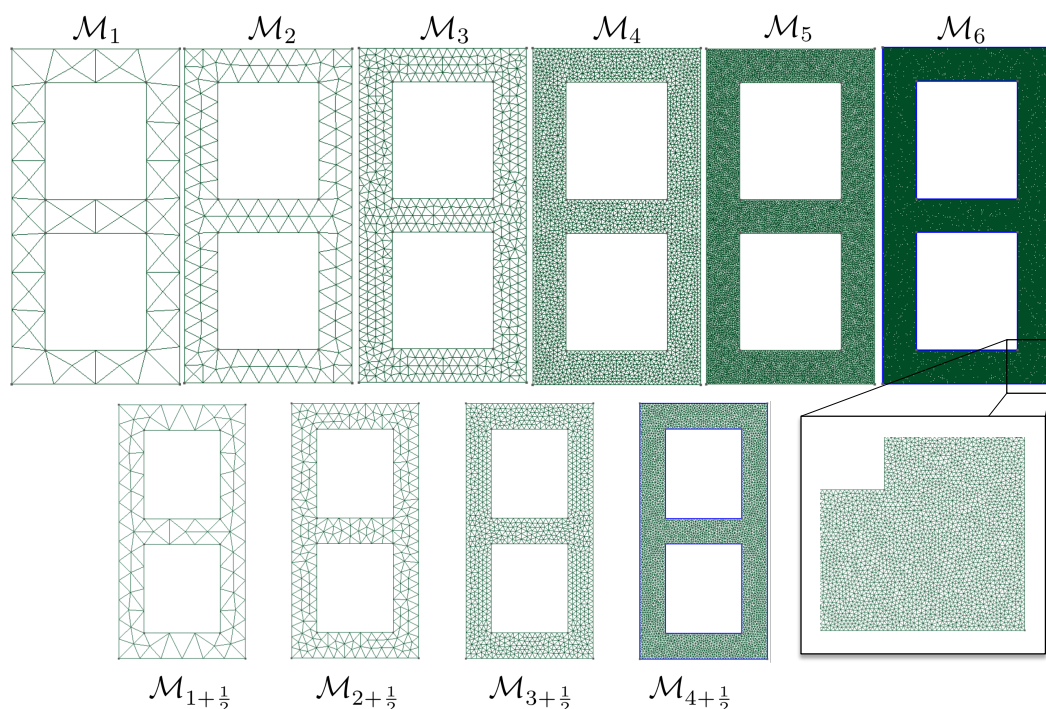


Figure 2. Hierarchy of computational meshes used in the study. We aim at selecting the coarsest mesh that delivers the targeted distributions up to a user-defined numerical accuracy. The second line of meshes is only used for error estimation purposes.

3.3. Inverse Problems and First Results

Two tests will now be investigated:

- Test 1 (weakly informative data): only the first eigenvalue is measured, i.e., d is scalar. This can be interpreted as a task of model updating, where new data is used to update an existing knowledge.

- Test 2 (strongly informative data): the first three eigenvalues of the structure are measured. This can be interpreted as an inverse problem, where rich information is used to identify all the unknown of the model, and the probabilistic setting acts as a regulariser.

The two corresponding marginal posterior density $\hat{\pi}_{\mu_1, \mu_3 | d}$ obtained when using mesh \mathcal{M}_3 are represented in Figure 3. Samples from this density are obtained by a Monte-Carlo sampler, as presented in next section. A Kernel Density Estimate (KDE) is used as a smoother for illustration purposes only. Notice that for the predictive posterior densities, the histograms correspond to the marginal densities of each individual eigenvalue, which explains their overlap.

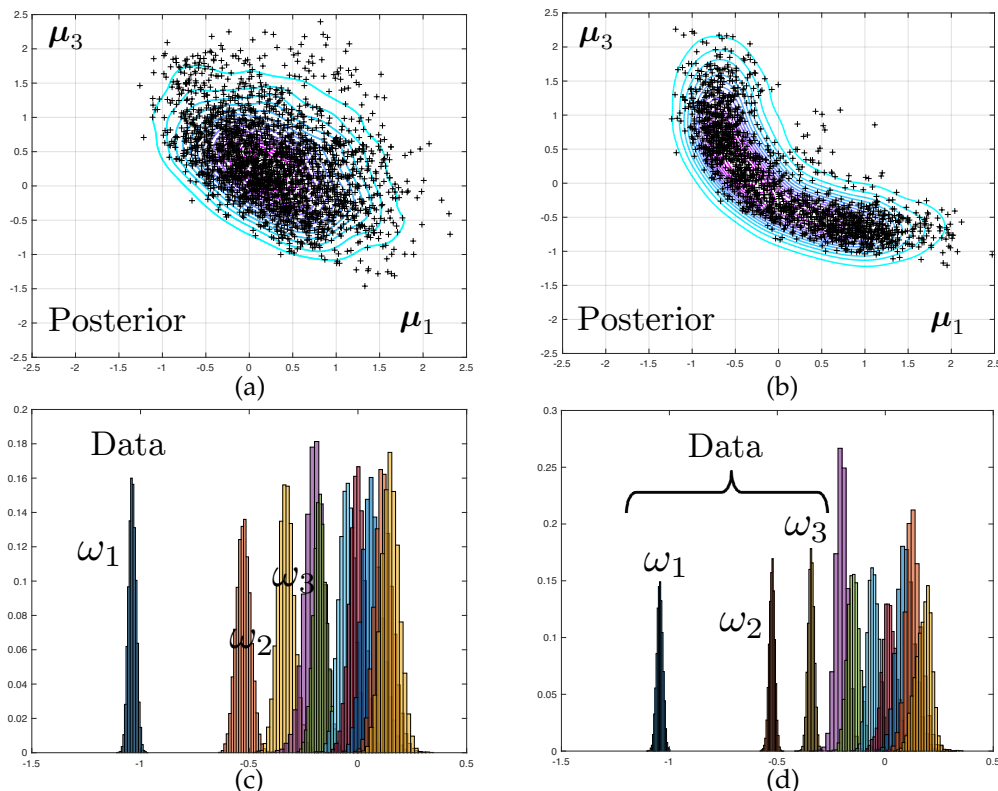


Figure 3. (a) Joint posterior probability distribution for two of the 6 unknown elastic moduli and (c) marginal posterior predictive distributions of the first 10 free vibration frequencies. The data set consists of a noisy measurement of the first frequency only, resulting in fat posterior densities. (b) Posterior distribution of the two unknown elastic moduli when the first 3 vibration frequencies are used as dataset, which results as in much thinner posterior densities. (d) Corresponding posterior predictive densities.

It is interesting to notice that the posterior densities observed in Test 2 are much sharper than that of Test 1, owing to the quality of the data. The symmetry in the results are a consequence of structural and probabilistic symmetries (see Figure 1 and the definition of the prior probability density).

Synthetic data for this problem is generated by computing the average of the spectra delivered by meshes \mathcal{M}_2 and $\mathcal{M}_{2+\frac{1}{2}}$, for a reference parameter vector $\mu_d \neq \mu_0$ but situated in the vicinity of the prior mode. The model averaging, together with selecting relatively coarse finite element meshes in sequence $\{\mathcal{M}_i\}_{i \in \llbracket n_m \rrbracket}$, is meant to circumvent the “inverse crime” problem.

3.4. Convergence with Mesh Refinement

The posterior densities corresponding to Test 2 and to various level of mesh refinement are displayed in Figure 4. The difference between the modes of the predicted eigenvalues that are obtained with coarse and fine meshes are very large. The fact that this discrepancy increases with the mode

number is to be expected as the spatial wave length of the deformations in the continuum can be shown to increase linearly with the eigenfrequency. As a consequence, a good mesh for the first range of frequencies might be unable to capture the faster spatial variations associated with higher vibration mode.

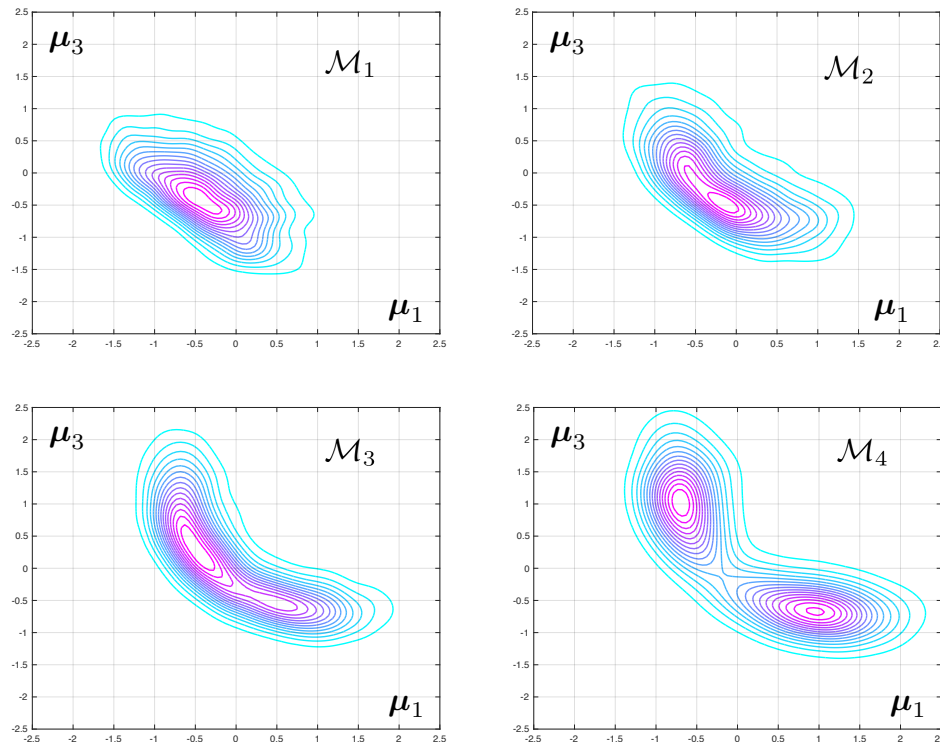


Figure 4. Evolution of the joint posterior density of two of the unknown elasticity parameters as the computational mesh is progressively refined from \mathcal{M}_1 to \mathcal{M}_4 .

The difference between the posteriori densities corresponding to meshes \mathcal{M}_3 and \mathcal{M}_4 is qualitatively small. The solution of the Bayesian inverse problem converges with mesh refinement. Notice that the posterior distribution for the model parameters goes from mono-modal to multi-modal, which may prove a stumbling block when selecting an appropriate Monte-Carlo sampler.

4. Monte-Carlo Sampler and Combined Effect of Statistical and Discretisation Errors

4.1. Tempered Metropolis-Hastings Markov Chain Monte Carlo Algorithm

The approximate posterior density described by Equation (11) is an arbitrarily complex function of μ , as it depends on the *nonlinear* computational model $(A_d(\mu)^T A_p(\mu)^T)^T$. In particular, standard random number generators cannot be used to draw samples from this distribution. Importance Monte-Carlo procedures whereby the prior distribution is used as proposal density, will also fail. This is because the posterior density may be arbitrarily different from the prior density, resulting in unacceptably large variances of importance sampling estimates. Designing better proposal densities a priori is impossible and a successful importance sampling approach would require generating the proposal density using advanced methods such as sequential Monte-Carlo samplers.

One of the simplest and generic generator of samples from posterior densities is the Metropolis-Hastings (MH) Markov Chain Monte Carlo (MCMC) algorithm (see Figure 5).

The algorithm works as follows. Starting from sample μ_n , the next sample μ_{n+1} is drawn from transitional distribution

$$T_{\mu_{n+1} \leftarrow \mu_n}(\mu_{n+1}; \mu_n) \propto \min \left(1, \frac{\bar{\pi}_{\mu|d}(\mu_{n+1})}{\bar{\pi}_{\mu|d}(\mu_n)} \right) \mathcal{N}(\mu_{n+1}; \mu_n, \tilde{\Sigma}) \quad (16)$$

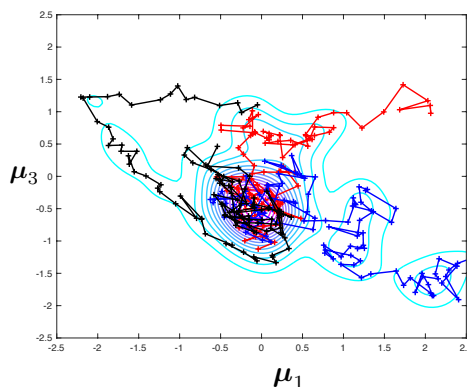


Figure 5. Three independent MCMC chains sampling a multivariate Gaussian distribution.

This is done by first drawing a random move from μ_k using the multivariate Gaussian (any proposal can be used, but the expressions exposed in this section are only valid for symmetric proposals), and then accepting or rejecting the move in order to account for the first term of the transition. Typically, a move ending up in a state of higher posterior density is always accepted, whilst a move ending up in a state of lower density may be accepted or not, depending on the result of a die and the ratio of posterior densities between current and proposed states.

This particular transition is designed such that the following ergodic property holds:

$$\int_{\mu_{n+1}} T_{\mu_{n+1} \leftarrow \mu_n}(\mu_{n+1}; \mu_n) d\mu_n = \bar{\pi}_{\mu|d}(\mu_{n+1}) \quad (17)$$

As a result, under some assumptions, the Markov chain is guaranteed to have $\bar{\pi}_{\mu|d}$ as its stationary distribution, which means that $\mu_n \sim \bar{\pi}_{\mu|d}$ as $n \rightarrow \infty$. Each sample, taken individually, is distributed according to the posterior density, if the chain is run long enough. Due to the Markov process, the samples are not independent, which makes frequentist error estimation and convergence diagnosis more difficult than in the context of traditional Monte-Carlo algorithm, where tools such as standard error and bootstrap apply without particular difficulty. Practically, a burn-in phase is first observed, whereby the chain “seeks” the regions of high probability densities (see Figure 5). Once found, samples become progressively distributed according to the target posterior density.

Notice that $\tilde{\Sigma}$ is difficult to choose a priori. Adaptive proposal MCMC have been proposed in [40,41]. Alternative solutions to choose good proposal are methods based on particle mechanics (e.g., Langevin diffusion, Hamiltonian dynamics) (see e.g., [42]). In this particular contribution, the proposal densities have been calibrated “by hand”. We use a tempered version of the MCMC algorithm, whereby we simultaneously sample multiple tempered replicates of the posterior density, with proposed state exchange between replicates that are subsequently Metropolis corrected [43,44] (see also [45] for an interesting application in the context of structural damage assessment). This relatively classical MCMC algorithm allows us to sample from; densities that are multi-modal or become multi-modal with mesh refinement (at least in the low-dimensional parametric settings that is investigated in this paper, as multi-modality in high-dimensions remains an open problem).

The call to the finite element solver is hidden in the acceptance/rejection test, which requires the evaluation of the finite element likelihood at the proposed state. Therefore, as for a standard MC algorithm, one iteration means one evaluation of the computational model.

4.2. Empirical Posterior Densities

Once n samples $\tilde{\mathcal{M}} = \{\mu_1, \mu_2, \dots, \mu_n\}$ have been computed (n larger than 10) by the MH algorithm, we discard the first 25% (burn-in) of these samples, resulting in \tilde{n} samples that we hope are located in the ergodic part of the Markov chain. The resulting empirical probability density is given by

$$\pi_{\mu|d}(\mu) \approx \tilde{\pi}_{\mu|d}^{(n)}(\mu) := \frac{1}{\tilde{n}} \sum_{k=n-\tilde{n}+1}^n \delta(\mu - \mu_k) \quad (18)$$

where δ is the Dirac delta function. Accordingly, the empirical predictive posterior density is

$$\pi_{p|d}(p) \approx \tilde{\pi}_{p|d}^{(n)}(p) := \frac{1}{\tilde{n}} \sum_{k=n-\tilde{n}+1}^n \delta(p - A_p(\mu_k)) \quad (19)$$

The cumulative empirical predictive posterior density may be expressed as

$$\bar{\Pi}_{\mu|d=d}(\mu) := \frac{1}{\tilde{n}} \sum_{k=n-\tilde{n}+1}^n \mathbb{I}(\mu; \mu_k \leq \mu) \quad (20)$$

where \mathbb{I} is the indicator function, and the inequality is to be understood in a component-by-component manner. A similar definition holds for the cumulative posterior density of p .

4.3. Total Error Measure

It is now clear that both the finite element discretisation and the Monte-Carlo approximate sampling affect the quality of the resulting posterior densities. The total error, for the marginal distribution of a single element μ_i of μ , reads as

$$\mathcal{D}_{\text{ks}}(\pi_{\mu_i|d}(\mu_i), \tilde{\pi}_{\mu_i|d}^{(n)}(\mu_i)) = \sup_{\mu_i} \left(\left| (\Pi_{\mu_i|d}(\mu_i) - \bar{\Pi}_{\mu_i|d}^{(n)}(\mu_i)) \right| \right) \quad (21)$$

which can be formally decomposed as follows

$$\begin{aligned} \mathcal{D}_{\text{ks}}(\pi_{\mu_i|d}(\mu_i), \tilde{\pi}_{\mu_i|d}^{(n)}(\mu_i)) &= \sup_{\mu_i} \left(\left| (\Pi_{\mu_i|d}(\mu_i) - \bar{\Pi}_{\mu_i|d}(\mu_i)) + \bar{\Pi}_{\mu_i|d}(\mu_i) - \bar{\Pi}_{\mu_i|d}^{(n)}(\mu_i) \right| \right) \\ &\leq \mathcal{D}_{\text{ks}}(\pi_{\mu_i|d}(\mu_i), \bar{\pi}_{\mu_i|d}(\mu_i)) + \mathcal{D}_{\text{ks}}(\bar{\pi}_{\mu_i|d}(\mu_i), \tilde{\pi}_{\mu_i|d}^{\text{mc}}(\mu_i)) \end{aligned} \quad (22)$$

In the last expression, the first term is the pure finite element error, which would occur if we could run the Markov process for an infinite number of iterations, while the second term is a pure statistical error.

We also define the error of an element p_i of p as follows:

$$\mathcal{D}_{\text{ks}}(\pi_{p_i|d}(p_i), \tilde{\pi}_{p_i|d}^{(n)}(p_i)) = \sup_{p_i} \left(\left| (\Pi_{p_i|d}(p_i) - \bar{\Pi}_{p_i|d}^{(n)}(p_i)) \right| \right) \quad (23)$$

5. Robust, Automatised and Comprehensive Error Control

5.1. Simulation of the Discretisation Error

The finite element method introduces an error in computational mapping $A(\mu) = \left(A_d(\mu)^T \ A_p(\mu)^T \right)^T$. Mapping A contains all the scalar quantities that need to be evaluated through calls to the finite element solver, namely the numerical predictions of the physical measurements d , and the posterior predictions p . We define the error in the simulated data as

$$\Delta A_d(\mu) = A_d(\mu) - \bar{A}_d(\mu) \quad (24)$$

and the error in the posteriori predictions as

$$\Delta A_p(\mu) = A_p(\mu) - \bar{A}_p(\mu) \quad (25)$$

For now, we assume that both these quantities can be estimated, at affordable numerical cost and in a reliable manner. Therefore, for any value of parameter μ , a corrected computational model is available, which reads as

$$\begin{pmatrix} A_d(\mu) \\ A_p(\mu) \end{pmatrix} \underset{\text{very close to}}{\approx} \begin{pmatrix} \hat{A}_d(\mu) \\ \hat{A}_p(\mu) \end{pmatrix} := \begin{pmatrix} \bar{A}_d(\mu) \\ \bar{A}_p(\mu) \end{pmatrix} + \begin{pmatrix} \widehat{\Delta A}_d(\mu) \\ \widehat{\Delta A}_p(\mu) \end{pmatrix} \quad (26)$$

where symbol $\hat{\cdot}$ denotes computable estimates.

5.2. Component-Wise MCMC

It is now possible to sample the corrected posterior distribution

$$\hat{\pi}_{\mu|d}(\mu; d) = \frac{\hat{\mathcal{L}}(\mu; d) \pi_{\mu}(\mu)}{\hat{\pi}(d)} \quad (27)$$

using MCMC. It should be clear that the corrected posterior distribution is simply obtained by replacing the corrected computational model (26) into the expression of the likelihood function, Equation (3). Notice that the normalising constant is affected by modifications of the computational model. This is of no practical consequence as MCMC samplers work with unnormalised densities, and the KS distance uses empirical cumulative distributions directly, without the need for smoothing or marginalisation.

We formally define a component-wise MCMC where the uncorrected and corrected computational models are sampled at the same time. This will yield an estimate of the effect of the discretisation error on posteriori densities at any stage of the Markov process, which, in turn, will allow us to develop an early-stopping methodology. The Component-wise MCMC iteration proceeds as follows, given a current sample $(\bar{\mu}_n, \hat{\mu}_n)$ of the uncorrected/corrected finite element posteriori densities,

- Draw $(\bar{\mu}_{n+1}, \hat{\mu}_{n+1})$ such that

$$\bar{\mu}_{n+1} \sim \mathcal{N}(\cdot; \bar{\mu}_n, \tilde{\Sigma}) \quad \text{and} \quad \hat{\mu}_{n+1} \sim \mathcal{N}(\cdot; \hat{\mu}_n, \tilde{\Sigma}) \quad (28)$$

- draw (u, v) such that

$$u \sim \mathcal{U}([0, 1]) \quad \text{and} \quad v \sim \mathcal{U}([0, 1]) \quad (29)$$

- Accept $\bar{\mu}_{n+1}$ if and only if

$$u \leq \min \left(1, \frac{\bar{\mathcal{L}}(\bar{\mu}_{n+1}; d) \pi_{\mu}(\bar{\mu}_{n+1})}{\bar{\mathcal{L}}(\bar{\mu}_n; d) \pi_{\mu}(\bar{\mu}_n)} \right) \quad (30)$$

set $\bar{\mu}_{n+1} = \bar{\mu}_k$ otherwise.

- Accept $\hat{\mu}_{n+1}$ if and only if

$$v \leq \min \left(1, \frac{\hat{\mathcal{L}}(\hat{\mu}_{n+1}; d) \pi_{\mu}(\hat{\mu}_{n+1})}{\hat{\mathcal{L}}(\hat{\mu}_n; d) \pi_{\mu}(\hat{\mu}_n)} \right) \quad (31)$$

set $\hat{\mu}_{n+1} = \hat{\mu}_k$ otherwise.

The MCMC algorithm is initialised by state $(\bar{\mu}_0, \hat{\mu}_0)$, where both $(\bar{\mu}_0$ and $\hat{\mu}_0)$ are drawn from distribution π_0 .

In the field of a posteriori finite element error estimation, the error estimate is usually a post-processing operation of the coarse finite element solution. This adapts seamlessly to

non-Markovian Monte-Carlo samplers, by post-processing each of the independent samples. Here, unfortunately, the finite element model has to be called twice at every iteration of the Markov process. Whether this can be avoided or not, for instance by making use of elements of sequential Monte-Carlo samplers, is unclear to us at this stage.

5.3. Machine Learning-Based Simulation of the Discretisation Error

At this stage, any numerical error estimator can be used, provided that it is goal-oriented. A method of choice could be a residual-based [5,6] or smoothing-based a posteriori error estimator [12,46] in conjunction with the adjoint methodology [8,9,11]. In addition, nothing prevents us from using a meta-modelling approach, such as projection-based reduced order modelling [47–52] or polynomial chaos expansions [16,17] to approximate the variations of the computed quantities of interest with parameter variation. Error estimates also exist for such two-level approximations.

In this contribution, we develop and use a feature-based method, that finds its roots in data-science methodologies and is much more “black-box” than the previously mentioned strategies. The proposed technique is inspired by the work of [35–37].

The exact continuum mechanics model is well approximated by a very refined numerical strategy (e.g., no meta-modelling and a very fine mesh). However, this very refined numerical model cannot be used at every iteration of MCMC as its evaluation is very costly. We propose to train a model that will map parameter μ to the output of the generally intractable very fine model through the combination of (i) a dedicated feature extractor and (ii) a weakly parametric regression model. This combination is defined as

$$\begin{pmatrix} A_d^*(\mu) \\ A_p^*(\mu) \end{pmatrix} - \begin{pmatrix} \bar{A}_d(\mu) \\ \bar{A}_p(\mu) \end{pmatrix} \underset{\text{wished}}{=} \begin{pmatrix} \widehat{\Delta A}_d(\mu) \\ \widehat{\Delta A}_p(\mu) \end{pmatrix} \underset{\text{constructed}}{=} \mathcal{R}(\mathcal{F}(\mu); \theta) \quad (32)$$

where the $*$ denote quantities that are delivered by the overkill (but computable) numerical model and \mathcal{R} is a weakly parametrised regression model, here a neural network regression (a Gaussian process could be used as well, but a random forest would probably have been the most efficient choice, given the way we bootstrap the regression model to generate estimates of generalisation errors), parametrised by a set of parameters $\theta \in \mathbb{R}^{n_\theta}$ (we do not explicitly distinguish parameters and hyper-parameters in our notations). \mathcal{F} is a mapping from input μ to a feature space. Its careful design is critical to the success of the machine learning procedure. We choose to construct the following features:

$$\mathcal{F}(\mu) = \begin{pmatrix} \tilde{A}_d(\mu) - \bar{A}_d(\mu) \\ \tilde{A}_d(\mu) \\ \tilde{A}_p(\mu) - \bar{A}_p(\mu) \\ \bar{A}_p(\mu) \end{pmatrix} \quad (33)$$

where $\tilde{A}_d(\mu)$ are *slightly* corrected computational models, here generated by refining the mesh by a moderate factor. In our examples, the typical mesh size is divided by 1.5 (see Figures 2 and 6 where we use mesh $\mathcal{M}_{i+\frac{1}{2}}$ to correct mesh \mathcal{M}_i , whilst the overkill solution is computed using \mathcal{M}_{i+2}). In this fashion, the generation of features will remain of the order of the computation of the finite element solution itself.

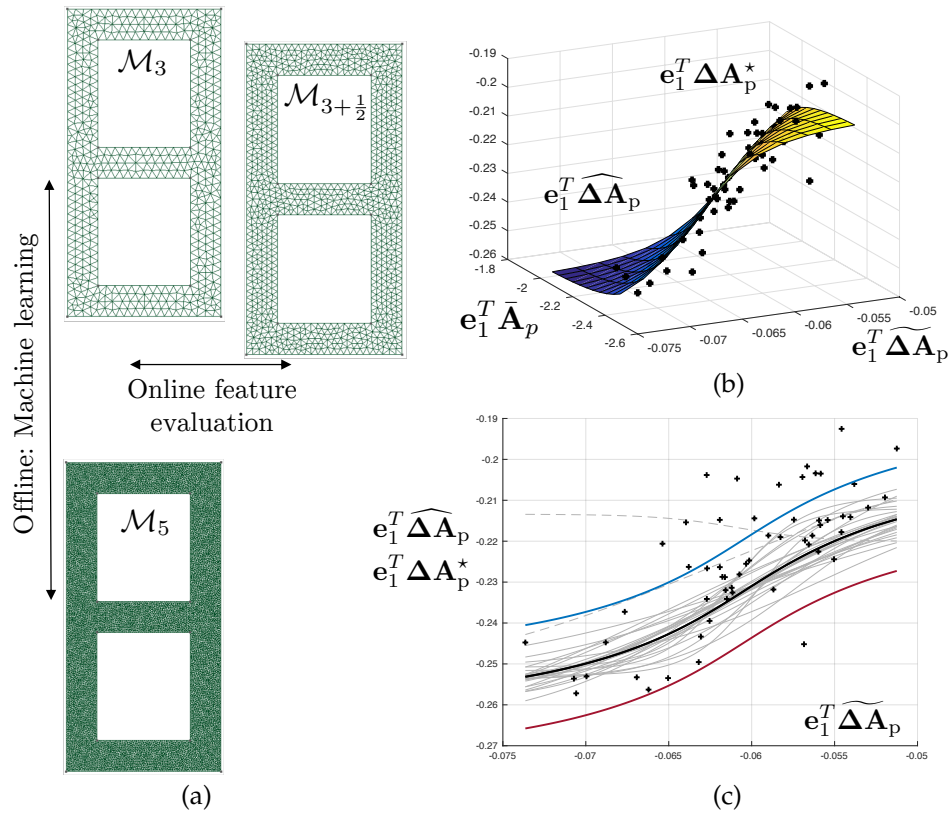


Figure 6. Machine-learning-based error estimation procedure. The overkill error (difference between the current mesh and a much finer mesh as shown in (a)) is postulated as being an unknown function of an error estimate. The inexpensive error estimate is the distance between the results obtained when using the current mesh and those obtained when using a slightly refined mesh as shown in. (b) The statistical learning is done offline via a Monte-Carlo sampling of the true error (using the prior as sampling density) and the adaptive fitting of a Neural Network regression. The regression is bootstrapped to provide enhanced stability, and derive confidence interval estimates for the overall error estimation procedure which is shown in (c). Both the size of the dataset and the hyperparameters of the network are found automatically via a Greedy process that is not detailed in this paper.

Now, we train $n_d + n_p$ multivariate neural network regression models

$$\begin{pmatrix} \widehat{\Delta A}_d(\mu) \\ \widehat{\Delta A}_p(\mu) \end{pmatrix} = \begin{pmatrix} \mathcal{R}_{d,1} \left(\begin{pmatrix} e_1^{n_d T} (\tilde{A}_d(\mu) - \bar{A}_d(\mu)) \\ e_1^{n_d T} \tilde{A}_d(\mu) \end{pmatrix}, \theta_{d,1} \right) \\ \dots \\ \mathcal{R}_{d,n_d} \left(\begin{pmatrix} e_{n_d}^{n_d T} (\tilde{A}_d(\mu) - \bar{A}_d(\mu)) \\ e_{n_d}^{n_d T} \tilde{A}_d(\mu) \end{pmatrix}, \theta_{d,n_d} \right) \\ \mathcal{R}_{p,1} \left(\begin{pmatrix} e_1^{n_p T} (\tilde{A}_p(\mu) - \bar{A}_p(\mu)) \\ e_1^{n_p T} \tilde{A}_p(\mu) \end{pmatrix}, \theta_{p,1} \right) \\ \dots \\ \mathcal{R}_{p,n_p} \left(\begin{pmatrix} e_{n_p}^{n_p T} (\tilde{A}_p(\mu) - \bar{A}_p(\mu)) \\ e_{n_p}^{n_p T} \tilde{A}_p(\mu) \end{pmatrix}, \theta_{p,n_p} \right) \end{pmatrix} \quad (34)$$

where e_j^m denotes the j th canonical vector of \mathbb{R}^m .

Each of the regression $\mathcal{R}_{l,i}$ is a single-hidden layer bootstrap-aggregated neural network model with n_n neurons and n_{nbs} bootstrap replicates.

$$\mathcal{R}_{l,i} \left(\begin{pmatrix} x \\ y \end{pmatrix}, \theta_{l,i} \right) = \frac{1}{k} \sum_{k=1}^{n_{nbs}} \sum_{j=1}^{n_n} a_j^{(k)} \tanh \left(a_{x,j}^{(k)} x + a_{y,j}^{(k)} y + o_j^{(k)} \right) + o^{(k)} \quad (35)$$

Training

We sample artificial “data” by running the overkill computational model n_{ml} times, after sampling the training set parameters $\{\mu_1^{ml}, \mu_2^{ml}, \dots, \mu_{n_{ml}}^{ml}\}$ from prior π_μ . The number of neurons and the cardinality of the training set is chosen automatically by making use of an automatised early-stopping methodology that aims to maximise the predictive coefficient of determination. We will not detail this procedure here. Fitting of the nonlinear regression coefficients is performed by employing standard least-squares method, solved by a gradient descent algorithm with randomised initialisation. Outliers of the set of bootstrap replicates are identified and eliminated to decrease the variance of the bootstrap-aggregated regression model.

An example of fitted regression model is represented in Figure 6, where the output is the discretisation error in the first free vibration circular frequency (i.e., regression model $\mathcal{R}_{d,1}$).

5.4. Bootstrap Confidence Intervals for the MCMC Sampler

At any iteration n of the MCMC sampler, a Monte-Carlo estimate of the discretisation error for the posterior density of the i th component of μ is given by

$$\mathcal{D}_{ks}^{(n)} := \mathcal{D}_{ks}(\hat{\pi}_{\mu_i|d}^{(n)}(\mu_i), \bar{\pi}_{\mu_i|d}^{(n)}(\mu_i)) = \sup_{\mu_i} \left(\left| (\hat{\Pi}_{\mu_i|d}^{(n)}(\mu_i) - \bar{\Pi}_{\mu_i|d}^{(n)}(\mu_i)) \right| \right) \quad (36)$$

where

$$\bar{\Pi}_{\mu_i|d}^{(n)}(\mu_i) := \frac{1}{\tilde{n}} \sum_{k=n-\tilde{n}+1}^n \mathbb{I}(\mu_i; \bar{\mu}_{k,i} \leq \mu_i), \quad (37)$$

and

$$\hat{\Pi}_{\mu_i|d}^{(n)}(\mu_i) := \frac{1}{\tilde{n}} \sum_{k=n-\tilde{n}+1}^n \mathbb{I}(\mu_i; \hat{\mu}_{k,i} \leq \mu_i). \quad (38)$$

Crucially, $\mathcal{D}_{ks}^{(n)}$ is a random variable whose statistics, and in particular its bias and variance, strongly depend on the length of the Markov chain. Unfortunately, evaluating the convergence of any statistics provided by MCMC is difficult, due to the statistical dependency between successively drawn samples.

Following standard diagnostic approaches for MCMC samplers (e.g., the Gelman-Rubin convergence test [53,54]), we will run $n_c \geq 10$ independent (tempered) MCMC chains in parallel and pool all the resulting samples, after discarding the first 25% of every individual chain as burn-in (see Figure 7 as a visual aid). The pooled samples at iteration n of the multiple-chain MCMC (MC³) algorithm are

$$\bar{\mathcal{S}}^{(n)} = \prod_{i=1}^{n_c} \bar{\mathcal{S}}_i^{(n)} \quad \hat{\mathcal{S}}^{(n)} = \prod_{i=1}^{n_c} \hat{\mathcal{S}}_i^{(n)} \quad (39)$$

where \prod denotes the cartesian product. In the previous expression, the sample set from chain i (at ambient temperature) is

$$\bar{\mathcal{S}}_i^{(n)} = \{\bar{\mu}_{i,n-\tilde{n}+1}, \dots, \bar{\mu}_{i,n}\} \quad \hat{\mathcal{S}}_i^{(n)} = \{\hat{\mu}_{i,n-\tilde{n}+1}, \dots, \hat{\mu}_{i,n}\} \quad (40)$$

$\mathcal{D}_{\text{ks}}^{(n)}$ is now the pooled KS distance estimate provided by the MC³ algorithm (we will keep the same notation for the sake of simplicity). Formally, we simply replace Equation (37) by

$$\bar{\Pi}_{\mu_i|d=d}^{(n)}(\mu_i) := \frac{1}{n_c} \frac{1}{\tilde{n}} \sum_{l=1}^{n_c} \sum_{k=n-\tilde{n}+1}^n \mathbb{I}(\mu_i; \bar{\mu}_{l,k,i} \leq \mu_i), \quad (41)$$

and perform a similar operation to define the pooled corrected empirical distribution.

The independence of the n_c MCMC chains allows us to compute confidence intervals for $\mathcal{D}_{\text{ks}}^{(n)}$ by making use of the non-parametric bootstrap. This is done by resampling $\mathcal{S}^{(n)}$ and $\hat{\mathcal{S}}^{(n)}$ with replacement, generating bootstrap replicates of the pooled sample sets $\{\bar{\mathcal{S}}_k^{(n)}\}_{k \in \llbracket 1 \ n_{\text{bs}} \rrbracket}$ and $\{\hat{\mathcal{S}}_k^{(n)}\}_{k \in \llbracket 1 \ n_{\text{bs}} \rrbracket}$,

$$\bar{\mathcal{S}}_k^{(n)} = \prod_{i \in \mathcal{B}_k} \bar{\mathcal{S}}_i^{(n)} \quad \hat{\mathcal{S}}_k^{(n)} = \prod_{i \in \mathcal{B}_k} \hat{\mathcal{S}}_i^{(n)}, \quad (42)$$

where $\mathcal{B}_k \in (\llbracket 1 \ n_c \rrbracket)^{n_{\text{bs}}}$ is such that each element of this set is drawn uniformly over $\llbracket 1 \ n_c \rrbracket$ and k varies between 1 and a large number n_{bs} , typically set to 1000. For each replicate, statistics $\mathcal{D}_{\text{ks}}^{(n)}$, denoted by $\mathcal{D}_{\text{ks},k}^{(n)}$, can be computed in a straightforward manner by using the bootstrap replicates of the pooled empirical distributions

$$\bar{\Pi}_{\mu_i|d=d,k}^{(n)}(\mu_i) := \frac{1}{n_c} \frac{1}{\tilde{n}} \sum_{\bar{\mu}_i^* \in \bar{\mathcal{S}}_k^{(n)}} \mathbb{I}(\mu_i; \bar{\mu}_i^* \leq \mu_i), \quad (43)$$

$$\hat{\Pi}_{\mu_i|d=d,k}^{(n)}(\mu_i) := \frac{1}{n_c} \frac{1}{\tilde{n}} \sum_{\hat{\mu}_i^* \in \hat{\mathcal{S}}_k^{(n)}} \mathbb{I}(\mu_i; \hat{\mu}_i^* \leq \mu_i), \quad (44)$$

which reads as

$$\mathcal{D}_{\text{ks},k}^{(n)} := \mathcal{D}_{\text{ks}}(\hat{\pi}_{\mu_i|d,k}^{(n)}(\mu_i), \bar{\pi}_{\mu_i|d,k}^{(n)}(\mu_i)) = \sup_{\mu_i} \left(\left| \hat{\Pi}_{\mu_i|d,k}^{(n)}(\mu_i) - \bar{\Pi}_{\mu_i|d,k}^{(n)}(\mu_i) \right| \right), \quad (45)$$

Finally, the bootstrap confidence intervals are calculated by calculating the X th and $(100 - X)$ th bootstrap percentiles such that the X th percentile reads as

$$q_X^{(n)} = \left(Q_X \left(\{\mathcal{D}_{\text{ks},k}^{(n)}\}_{k \in \llbracket 1 \ n_{\text{bs}} \rrbracket} \right) - \text{median} \left(\{\mathcal{D}_{\text{ks},k}^{(n)}\}_{k \in \llbracket 1 \ n_{\text{bs}} \rrbracket} \right) \right) + \mathcal{D}_{\text{ks}}^{(n)} \quad (46)$$

where Q_X is an operator that extracts the X th and Y th percentile of the set passed as argument.

It is important to understand that the derived bootstrap confidence interval stands for a chain of finite length n , and not for the asymptotic limit. In fact, estimate $\mathcal{D}_{\text{ks}}^{(n)}$ of \mathcal{D}_{ks} is strongly biased (upward for small asymptotic errors \mathcal{D}_{ks} and typically downward for large asymptotic errors), which is due to two factors:

- the existence of the burn-in phase. For small n , each individual chain will be strongly affected by the initialisation of the chains. For small asymptotic errors, this can be expected to have a strong upward bias effect providing that the initialisation is disperse, which is often the case in practice (e.g., initialisation from the prior, sequential MC approaches with decreasing levels of noise). This can be visualised in Figure 7. Two incomplete chains running on the same probability density may be exploring completely different regions of space, yielding values of \mathcal{D}_{ks} that are large, even in the case where the corrected and uncorrected densities are close to one another.
- the discrete evaluation of the KS distance itself, which generates an additional (upward) bias.

The variance of $\mathcal{D}_{\text{ks}}^{(n)}$ decreases with the number of chains of the MC³ algorithm (which is not a free parameter as overall CPU cost increases linearly with n_c). Of course, bias and variance are both expected to decrease with the length n of the run.

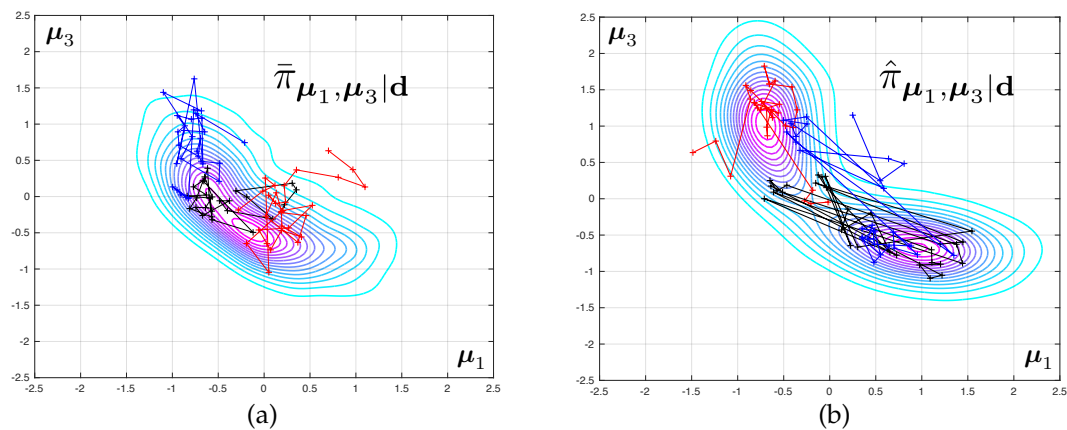


Figure 7. Multiple MCMC routines were run in parallel, for both the coarse likelihood function (coarse mesh) as shown in (a) and the enhanced likelihood function obtained after correction by the goal-oriented finite element error estimate, as shown in (b). The results are pooled, and will be bootstrapped to derive confidence intervals for the summary statistics of the posterior densities.

5.5. Simultaneous Control of All Sources of Errors

We now make use of the CI derived for $\mathcal{D}_{\text{ks}}^{(n)}$ to construct an adaptive inverse problem solver that jointly controls the quality of the mesh, and that of the statistical evaluation of the posterior densities. The algorithm is as follows. Given a current mesh \mathcal{M}_i and number of iterations n of the Monte-Carlo solver, do:

- If $n = 0$, perform m_0 iterations of the MC³ algorithm and set $n \leftarrow n + m_0$.
- Evaluate mesh convergence criterion \mathcal{C}_m . If this criterion is satisfied, exit the adaptation procedure;
- If \mathcal{C}_m is not satisfied, evaluate statistical convergence criterion \mathcal{C}_s
 - If \mathcal{C}_s is satisfied, set $i \leftarrow i + 1$, reinitialise the MC³ algorithm and set $n = 0$;
 - otherwise perform $m(n)$ iterations of the MC³ algorithm and set $n \leftarrow n + m(n)$ (m should be an exponentially increasing function).

5.5.1. Criterion \mathcal{C}_m

Convergence of the finite element-based Bayesian inverse problem is achieved when $\mathcal{D}_{\text{ks}} \leq \gamma$, where γ is a numerical tolerance that will typically be chosen between 0.01 and 0.2. As we only have Monte-Carlo estimates of \mathcal{D}_{ks} , we require instead that

$$\mathcal{C}_m : \quad q_{100-X}^{(n)} \leq \gamma \quad (47)$$

Criterion \mathcal{C}_m is an indicator of the combined effect of the discretisation and statistical errors onto the posterior densities. The discretisation error is evaluated through the choice of measure \mathcal{D}_{ks} as reliability indicator, whilst the statistical error is taken into account by making use of the upper limit of the bootstrap confidence interval for $\mathcal{D}_{\text{ks}}^{(n)}$. The risk of falsely detecting mesh convergence due to a high statistical error is small, due to the fact that $\mathcal{D}_{\text{ks}}^{(n)}$ is an upwardly biased estimate of \mathcal{D}_{ks} for small \mathcal{D}_{ks} . Confidence in the result can be increased through a loose Gelman-Rubin convergence test in order to eliminate the risk of stopping the MC³ algorithm in its non-ergodic phase. However, this has proved to be unnecessary in our numerical tests. This is because \mathcal{C}_m is a rather strict criterion.

5.5.2. Criterion \mathcal{C}_s

The second criterion will help us determine whether mesh refinement is actually needed, or whether the statistical error is too large for use to take a robust decision regarding mesh refinement. The ideal criterion is $\mathcal{D}_{\text{ks}} \geq \eta$, where $\eta < \gamma$. The obvious strategy that would require $q_X^{(n)} \leq \eta$ does not

work. This is due to the previously explained upward bias of estimator $\mathcal{D}_{\text{ks}}^{(n)}$, which would eventually lead to the spurious satisfaction \mathcal{C}_s , and consequently to systematic mesh refinement operations for low n count even when the mesh becomes fine enough for \mathcal{D}_{ks} to be well below target γ . In order to derive an appropriate criterion \mathcal{C}_s , we remark that whilst the ideal criterion $\mathcal{D}_{\text{ks}} \geq \eta$ cannot be statistically evaluated for general values of η , criterion $\mathcal{D}_{\text{ks}} > 0$ can be. More precisely, we propose to estimate the statistical error by evaluation whether, at current n , the corrected and uncorrected posterior distributions can be statistically distinguished.

We postulate the following null hypothesis: Hypothesis 0 (H0): *the corrected and uncorrected posterior densities are identical.*

The rejection of the null hypothesis will indicate that the two densities are significantly different. Now, the criterion for the need for mesh refinement becomes the following:

$$\mathcal{C}_s : \quad \Pr \left(\mathcal{D}_{\text{ks}}^{(n)} \geq q_Z^{(n)} | \text{H0} \right) \leq \xi \quad (48)$$

The probability density of $\mathcal{D}_{\text{ks}}^{(n)}$ under null hypothesis H0 can be approximated by adapting the previously described bootstrap procedure. We will estimate the density of

$$\mathcal{D}_{\text{ks},0}^{(n)} := \mathcal{D}_{\text{ks}}(\bar{\pi}_{\mu_i|d}^{(n),1}(\mu_i), \bar{\pi}_{\mu_i|d}^{(n),2}(\mu_i)) = \sup_{\mu_i} \left(\left| (\bar{\Pi}_{\mu_i|d}^{(n),1}(\mu_i) - \bar{\Pi}_{\mu_i|d}^{(n),2}(\mu_i)) \right| \right) \quad (49)$$

where $\bar{\pi}_{\mu_i|d}^{(n),1}$ and $\bar{\pi}_{\mu_i|d}^{(n),2}$ are obtained, respectively, by pooling the result of two independent runs of n_c Markov chains of lengths n and corresponding to the uncorrected finite element model only (one can equally choose the corrected one). Notice, to clarify the idea, that $\mathcal{D}_{\text{ks},0}^{(n)}$ tends to 0 as n tends to infinity: this is a measure of the statistical error only. The desired density can be estimated by resampling $\mathcal{S}^{(n)}$ twice, computing the KS distance between the two pooled sets of samples, and repeating the operation n_{bs} times, which generates a sequence of real numbers $\{\mathcal{D}_{\text{ks},0,k}^{(n)}\}$. The replicated empirical cumulative distributions are

$$\bar{\Pi}_{\mu_i|d=d,k}^{(n),j}(\mu_i) := \frac{1}{n_c} \frac{1}{\tilde{n}} \sum_{\bar{\mu}_i^* \in \mathcal{S}_k^{(n),j}} \mathbb{I}(\mu_i; \bar{\mu}_i^* \leq \mu_i) \quad (50)$$

with the sampling sets

$$\mathcal{S}_k^{(n),j} = \prod_{i \in \mathcal{B}_k^j} \mathcal{S}_i^{(n)} \quad (51)$$

where $j = 1$ or $j = 2$ and \mathcal{B}_k^1 and \mathcal{B}_k^2 are sets of $\llbracket 1 \ n_c \rrbracket^{n_c}$ constructed such that each element of these two sets is drawn uniformly over $\llbracket 1 \ n_c \rrbracket$. We can now extract the $(100 \times (1 - \xi))$ th percentile $q_{Y,0}^{(n)}$ corresponding to p-value ξ , and evaluate whether $q_{Y,0}^{(n)} \leq q_Y^{(n)}$. If so, test \mathcal{C}_s is true. If not, the statistical sampling error is too large to allow us to decide whether mesh refinement should be performed or not. In this case, and assuming that mesh convergence criterion \mathcal{C}_m is not satisfied, we need to continue sampling with MCMC.

Notice that the use of a larger percentile Z yields a less conservative indicator for the need for mesh refinement, which can be compensated by decreasing p-value ξ . The criterion that is arguably the easiest to interpret is $\Pr \left(\mathcal{D}_{\text{ks}}^{(n)} \geq q_{50}^{(n)} | \text{H0} \right) \leq \xi$, where $q_{50}^{(n)}$ is directly the KS distance $\mathcal{D}_{\text{ks}}^{(n)}$ (i.e., computed without bootstrapping).

6. Numerical Examples—Part II. Automatised Error Control and Discussion

We now come back to the numerical examples introduced in Section 3 and produce three series of results, illustrating three difference aspects of the proposed approach.

6.1. MCMC Iterations Only When Needed

The results reported in Figure 8 correspond to Test 1. For now, we aim to control the quality of the posterior density of the first parameter μ_1 . Each of the graphs shows the evolution of various statistics of the corresponding KS distance as a function of the number of iteration n of the tempered MC³ algorithm. The mesh used is displayed on the left-hand side of each of the graphs. The solid black line represents the evolution $\mathcal{D}_{\text{ks}}^{(n)}$ itself. The 5th and 95th bootstrap percentiles of this quantity are also reported, in solid blue line and solid red line, respectively. We set convergence criterion \mathcal{C}_m such that $\gamma = 0.2$ and $Y = 95$ for the pure discretisation error. Therefore, convergence with respect to the finite element discretisation is obtained when, for sufficient long chains, the red line (upper limit of the bootstrap CI for $\mathcal{D}_{\text{ks}}^{(n)}$) is below γ .

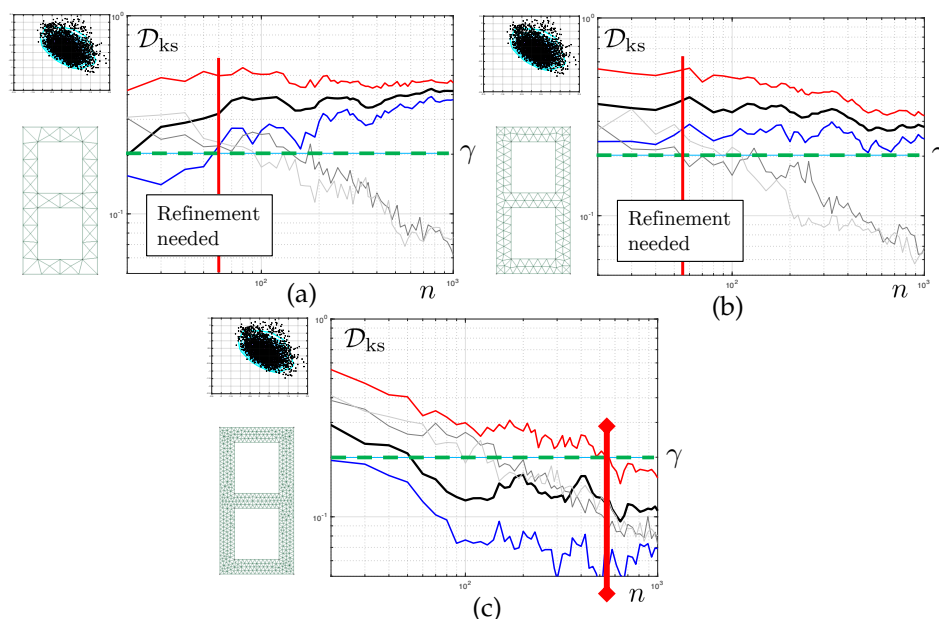


Figure 8. Convergence of the KS distance as a function of the number of Monte-Carlo samples in each chain (black line, with confidence interval represented by the area between the blue and red lines). The mesh used is represented schematically on the left-hand side of each of the graphs in (a–c). The data is composed of a unique noisy measurement of the first eigenfrequency. The quantity of interest is here the first elasticity parameter, in log scale. The grey lines are an indicator of the bias of the KS distance estimate, which decreases with the sample size. The bias should be small for the estimate to be robust, but not too small so as to avoid running unnecessarily long chains using meshes that are too coarse for our needs. Sufficiently accurate results are obtained with the third mesh. Importantly, very few MCMC iterations are needed to determine that the two first meshes are too coarse for our needs.

In grey colours, we report evolution of $\mathcal{D}_{\text{ks},0}^{(n)}$ for the coarse scale (light grey), consistently with the derivations of the previous section, but also, for reference, a similar statistics constructed with the corrected finite element model (dark grey). Both curves have similar evolutions and, of course, tend to 0 as n tends to infinity. We set $Z = 5$, and $\eta = 0.05$. This means that the statistical error is considered small enough once the blue line (lower limit of the bootstrap CI for $\mathcal{D}_{\text{ks}}^{(n)}$) is above the 95th percentile of $\mathcal{D}_{\text{ks},0}^{(n)}$. Here, “small enough” is to be understood, as small enough for a decision regarding the need to refine the mesh one step further to be taken.

The results are as follows. For the coarsest mesh, we detect a separation of the discretisation and statistical errors at iteration 60 (vertical solid red line). \mathcal{C}_s is satisfied. As \mathcal{C}_m is not satisfied, we can stop the MCMC sampler and refine the mesh. A similar behaviour is seen for mesh \mathcal{M}_2 . For the last

mesh, convergence is achieved after 530 iterations, \mathcal{C}_s never reaching satisfaction. The discretisation and sampling errors remain entangled, but the sum of them is below the desired target.

We can see here that the proposed algorithm allows us to stop the MCMC sampler early, moving directly to the level of mesh refinement that will yield the desired quality of the posterior densities.

6.2. Goal-Oriented Error Control

The second set of results correspond to Test 1, still. However, we now monitor the convergence of the posterior predictive density of the fourth eigenvalue. This is reported in the top two graphs of Figure 9. For \mathcal{M}_1 Early-stopping is performed at iteration 40 of the MCMC sampler. For \mathcal{M}_2 , global convergence is achieved after 120 iterations, and this error keeps decreasing. The MCMC is allowed to continue generating samples, meaning that the discretisation error is actually a lot smaller than the requested tolerance γ . It is interesting to see that the convergence is much faster than for the first parameter, whose convergence was studied in the previous subsection. This shows that for the same inverse problem, the algorithm may spend more or less resources depending on the engineering quantity of interest.

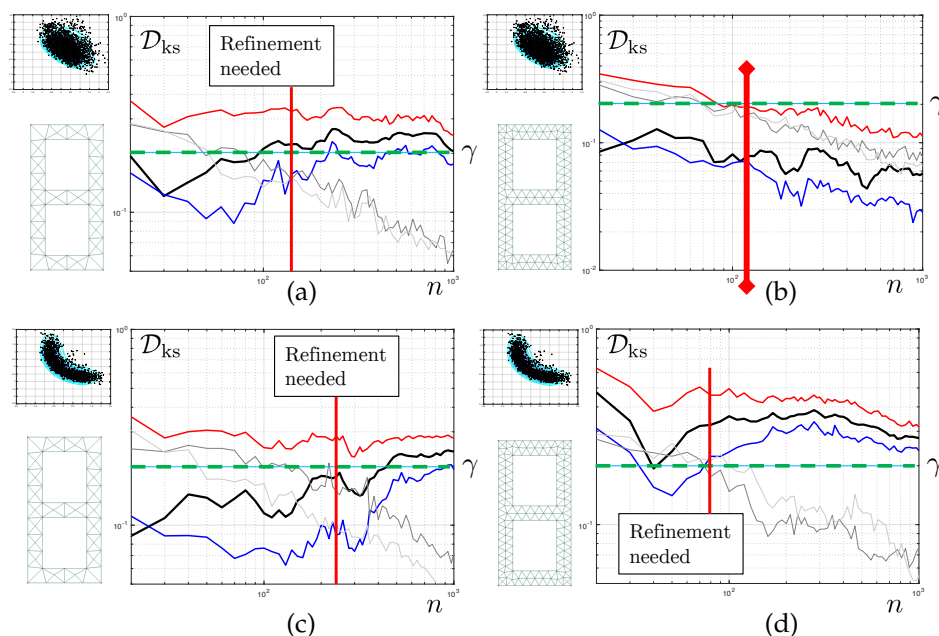


Figure 9. Convergence of the KS distance as a function of the number of Monte-Carlo samples in each chain. The (a,b) correspond to the inverse problem where only the first eigenfrequency is observed. (c,d) corresponds to the case where the first three eigenfrequencies are observed. The quantity of interest is the fourth eigenvalue, which is unobserved in both cases and must be inferred. The corresponding posterior density is illustrated above the mesh pictogram. Interestingly, the data-poor problem (a,b) can be solved appropriately with a coarse mesh, while the data-rich case (c,d) requires a much finer grid to be solved accurately.

6.3. Uncertainty-Driven Error Control

Finally, the last set of results concerns Test 2. Here, remember that the three first eigenvalues are used as measurements. We control the convergence of the fourth eigenvalue, which was controlled in the context of Test 1 in the previous subsection. We see here that for mesh \mathcal{M}_2 , the posterior density of the fourth eigenvalue is still far from convergence, whilst it was evaluated in a very precise way in Test 1. This shows that the proposed error control algorithm automatically adapts to the level of posterior uncertainty. Qualitatively, a wide posterior density is associated with a high uncertainty concerning the value of QoIs. Consequently, the mesh does not need to be very refined to capture the

posterior density correctly. Conversely, for Test 2, the posterior density is sharper, and similar levels of discretisation errors have a much stronger impact on the KS distance.

7. Concluding Remarks and Discussion

We have presented a methodology to control the various sources of errors arising in finite-element based Bayesian inverse problems. We have focussed on a simple numerical approximation chain consisting of (i) a finite element discretisation of the continuum mechanics problem and (ii) a MCMC solver to draw samples from posterior density distributions. So far we did not consider further error sources such as that engendered by meta-modelling. We have showed that it was possible to drive the mesh refinement process in a goal-oriented manner, by quantifying the impact of the associated error onto posterior density distributions. In order to do so, we run two independent MCMC simultaneously, one of them using a corrected likelihood function that takes the discretisation error. Any a posteriori error estimate available for the PDE under consideration may be used to obtain this correction. Of course, the accuracy (effectivity) of the chosen error estimate will impact the accuracy of the methodology developed in this paper. The study and control of this effect will require further research to be carried out,

We have shown that by using multiple replicates of the component-wise MCMC, it was also possible to (i) derive bootstrap-based confidence intervals for the error in posterior densities engendered by the spatial discretisation of the underlying PDE. Importantly, this approach allows us to stop the MCMC iteration as soon as we can be sufficiently confident in the fact that the posterior densities obtained with the current mesh are not accurate enough for our purpose. The approach is goal-oriented: the adaptivity is performed in the sense of the posterior distribution of either some of the latent parameters, or in the sense of the predictive posterior density of engineering QoIs. Finally, we have shown that the approach is uncertainty driven: it will only refine the mesh and/or request additional samples to be drawn by the MCMC algorithm if the effect on the QoIs can be felt when measured using a statistical distance between their posterior densities. In particular, model updating problems with wider priors tend to require less computational effort than parameter estimation problems with very rich observed data and/or narrow prior densities. The proposed methodology establishes a bridge between the field of reliability estimation for the sampling-based algorithms used to solve probabilistic inverse problems, and the field of deterministic, goal-oriented finite element error estimation.

Although the results presented in this paper are encouraging, the proposed approach has shortcomings that need to be addressed in future research work. The error estimation procedure is relatively wasteful for two reasons. Firstly, multiple MCMC algorithms need to run independently for bootstrapping to be possible. These chains all exhibit their own burn-in phase, which increases the amount of discarded samples. Secondly, the finite element error estimation procedure is not merely a post-processing operation any longer; one cannot post-process the finite element results corresponding to the uncorrected chain in order to compute the corrected likelihood. This is because both chains are independent and require running the coarse finite element simulations at different points of the parameter domain. Finally, let us acknowledge that Bayesian finite element inverse problems should not be solved by a MCMC solver without constructing a surrogate model first. The number of finite element computations involved, even if the proposal distribution is well designed, is in the tens of thousands. We are currently investigating the use of Polynomial Chaos surrogates, which adds another layer of numerical approximation that needs to be controlled in a robust and efficient manner. In this context, an elegant approach to separate the sources of errors (i.e., Finite Element discretisation error and Polynomial Chaos error) may be found in [55], which may constitute a solid starting point for the next step of our investigations.

Author Contributions: Methodology, P.K., A.K. and S.C.; Formal Analysis, P.K., A.K. and S.C.; Writing—Original Draft Preparation, P.K.; Writing—Review & Editing, A.K. and S.C.

Funding: We acknowledge the support provided by the Welsh Government and Higher Education Funding Council for Wales through the Sêr Cymru National Research Network in Advanced Engineering and Materials.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nagel, J.B.; Sudret, B. A Unified Framework for Multilevel Uncertainty Quantification in Bayesian Inverse Problems. *Probab. Eng. Mech.* **2016**, *43*, 68–84. [\[CrossRef\]](#)
2. Beck, J.L.; Au, S.K. Bayesian Updating of Structural Models and Reliability Using Markov Chain Monte Carlo Simulation. *J. Eng. Mech.* **2002**, *128*, 380–391. [\[CrossRef\]](#)
3. Chen, P.; Schwab, C. Sparse-Grid, Reduced-Basis Bayesian Inversion: Nonaffine-Parametric Nonlinear Equations. *J. Comput. Phys.* **2016**, *316*, 470–503. [\[CrossRef\]](#)
4. Cotter, S.L.; Dashti, M.; Stuart, A.M. Approximation of Bayesian Inverse Problems for PDEs. *SIAM J. Numer. Anal.* **2010**, *48*, 322–345. [\[CrossRef\]](#)
5. Ainsworth, M.; Oden, J. *A Posteriori Error Estimation in Finite Element Analysis*; Wiley: Chichester, UK, 2000.
6. Ladevèze, P.; Pelle, J.P. *Mastering Calculations in Linear and Non Linear Mechanics*; Springer: New York, NY, USA, 2004.
7. Díez, P.; Parés, N.; Huerta, A. *Encyclopedia of Aerospace Engineering*; Chapter Error Estimation and Quality Control; Wiley: Chichester, UK, 2010.
8. Oden, J.T.; Prudhomme, S.; Oden, J.T.; Prudhomme, S. Goal-Oriented Error Estimation and Adaptivity for the Finite Element Method. *Comput. Methods Appl. Mech. Eng.* **1999**, *41*, 735–756. [\[CrossRef\]](#)
9. Cirak, F.; Ramm, E. A Posteriori Error Estimation and Adaptivity for Elastoplasticity Using the Reciprocal Theorem. *Int. J. Numer. Methods Eng.* **2000**, *47*, 379–393. [\[CrossRef\]](#)
10. Strouboulis, T.; Babuška, I.; Datta, D.K.; Copps, K.; Gangaraj, S.K. A Posteriori Estimation and Adaptive Control of the Error in the Quantity of Interest. Part I: A Posteriori Estimation of the Error in the von Mises Stress and the Stress Intensity Factor. *Comput. Methods Appl. Mech. Eng.* **2000**, *181*, 261–294. [\[CrossRef\]](#)
11. Becker, R.; Rannacher, R. An Optimal Control Approach to a Posteriori Error Estimation in Finite Element Methods. *Acta Numer.* **2001**, *10*, 1–102. [\[CrossRef\]](#)
12. González-Estrada, O.; Nadal, E.; Ródenas, J.; Kerfriden, P.; Bordas, S.A.; Fuenmayor, F. Mesh Adaptivity Driven by Goal-Oriented Locally Equilibrated Superconvergent Patch Recovery. *Comput. Mech.* **2013**, *53*, 957–976. [\[CrossRef\]](#)
13. Jensen, H.; Esse, C.; Araya, V.; Papadimitriou, C. Implementation of an Adaptive Meta-Model for Bayesian Finite Element Model Updating in Time Domain. *Reliab. Eng. Syst. Saf.* **2017**, *160*, 174–190. [\[CrossRef\]](#)
14. Au, S.K.; Zhang, F.L. Fundamental Two-Stage Formulation for Bayesian System Identification, Part I: General Theory. *Mech. Syst. Signal Process.* **2016**, *66–67*, 31–42. [\[CrossRef\]](#)
15. Zhang, F.L.; Au, S.K. Fundamental Two-Stage Formulation for Bayesian System Identification, Part II: Application to Ambient Vibration Data. *Mech. Syst. Signal Process.* **2016**, *66–67*, 43–61. [\[CrossRef\]](#)
16. Babuška, I.; Tempone, R.; Zouraris, G.E. Galerkin Finite Element Approximations of Stochastic Elliptic Partial Differential Equations. *SIAM J. Numer. Anal.* **2004**, *42*, {800–825}. [\[CrossRef\]](#)
17. Ghanem, R.G.; Spanos, P.D. *Stochastic Finite Elements: A Spectral Approach*; Dover Publications: New York, NY, USA, 2003.
18. Nouy, A. Recent Developments in Spectral Stochastic Methods for the Numerical Solution Ofstochastic Partial Differential Equations. *Arch. Comput. Methods Eng.* **2009**, *16*, 251–285. [\[CrossRef\]](#)
19. Kundu, A.; DiazDelaO, F.; Adhikari, S.; Friswell, M. A Hybrid Spectral and Metamodeling Approach for the Stochastic Finite Element Analysis of Structural Dynamic Systems. *Comput. Methods Appl. Mech. Eng.* **2014**, *270*, 201–219. [\[CrossRef\]](#)
20. Ganapathysubramanian, B.; Zabarar, N. Sparse Grid Collocation Schemes for Stochastic Natural Convection Problems. *J. Comput. Phys.* **2007**, *225*, {652–685}. [\[CrossRef\]](#)
21. Foo, J.; Karniadakis, G.E. Multi-Element Probabilistic Collocation Method in High Dimensions. *J. Comput. Phys.* **2010**, *229*, 1536–1557. [\[CrossRef\]](#)

22. Pradlwarter, H. J.; Schuëller, G.I. On Advanced Monte Carlo Simulation Procedures in Stochastic Structural Dynamics. *Int. J. Non-Linear Mech.* **1997**, *32*, 735–744. [[CrossRef](#)]
23. Yamazaki, F.; Shinozuka, M. Digital Generation of Non-Gaussian Stochastic Fields. *J. Eng. Mech.* **1988**, *114*, 1183–1197. [[CrossRef](#)]
24. Au, S.K.; Beck, J.L. A New Adaptive Importance Sampling Scheme for Reliability Calculations. *Struct. Saf.* **1999**, *21*, 135–158. [[CrossRef](#)]
25. Rosenblueth, E. Point Estimates for Probability Moments. *Proc. Natl. Acad. Sci. USA* **1975**, *72*, 3812–3814. [[CrossRef](#)] [[PubMed](#)]
26. Christian, J.T.; Baecher, G.B. The Point-estimate Method with Large Numbers of Variables. *Int. J. Numer. Anal. Methods Geomech.* **2002**, *26*, 1515–1529. [[CrossRef](#)]
27. Julier, S.J. The Scaled Unscented Transformation. In Proceedings of the 2002 American Control Conference, Anchorage, AK, USA, 2002; Volume 6, pp. 4555–4559.
28. Yuen, K.V.; Kuok, S.C. Bayesian Methods for Updating Dynamic Models. *Appl. Mech. Rev.* **2011**, *64*, 010802. [[CrossRef](#)]
29. Cui, T.; Marzouk, Y.; Willcox, K. Scalable Posterior Approximations for Large-Scale Bayesian Inverse Problems via Likelihood-Informed Parameter and State Reduction. *J. Comput. Phys.* **2016**, *315*, 363–387. [[CrossRef](#)]
30. Kundu, A.; Matthies, H.G.; Friswell, M.I. Probabilistic optimization of engineering system with prescribed target design in a reduced parameter space. *Comput. Methods Appl. Mech. Eng.* **2018**, *337*, 281–304. [[CrossRef](#)]
31. Schillings, C.; Schwab, C. Sparse, Adaptive Smolyak Quadratures for Bayesian Inverse Problems. *Inverse Probl.* **2013**, *29*, 065011. [[CrossRef](#)]
32. Chen, P.; Schwab, C. Adaptive Sparse Grid Model Order Reduction for Fast Bayesian Estimation and Inversion. In *Sparse Grids and Applications—Stuttgart 2014*; Garcke, J., Pflüger, D., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 109, pp. 1–27.
33. Mattis, S.A.; Wohlmuth, B. Goal-Oriented Adaptive Surrogate Construction for Stochastic Inversion. *Comput. Methods Appl. Mech. Eng.* **2018**, *339*, 36–60. [[CrossRef](#)]
34. Pares, N.; Diez, P.; Huerta, A. Subdomain-Based Flux-Free a Posteriori Error Estimators. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 297–323. [[CrossRef](#)]
35. Drohmann, M.; Carlberg, K. The ROMES Method for Statistical Modeling of Reduced-Order-Model Error. *SIAM/ASA J. Uncertain. Quantif.* **2015**, *3*, 116–145. [[CrossRef](#)]
36. Paul-Dubois-Taine, A.; Amsallem, D. An Adaptive and Efficient Greedy Procedure for the Optimal Training of Parametric Reduced-Order Models. *Int. J. Numer. Methods Eng.* **2014**, *102*, 1262–1292. [[CrossRef](#)]
37. Gourey, O.; Amsallem, D.; Bordas, S.P.A.; Liu, W.K.; Kerfriden, P. Automatised Selection of Load Paths to Construct Reduced-Order Models in Computational Damage Micromechanics: From Dissipation-Driven Random Selection to Bayesian Optimization. *Comput. Mech.* **2016**, *58*, 213–234. [[CrossRef](#)]
38. Trehan, S.; Carlberg, K.T.; Durlofsky, L.J. Error Modeling for Surrogates of Dynamical Systems Using Machine Learning. *Int. J. Numer. Methods Eng.* **2017**, *112*, 1801–1827. [[CrossRef](#)]
39. Ciarlet, P.G. *The Finite Element Method for Elliptic Problems*; SIAM: Philadelphia, PA, USA, 1978.
40. Haario, H.; Laine, M.; Mira, A.; Saksman, E. DRAM: Efficient Adaptive MCMC. *Stat. Comput.* **2006**, *16*, 339–354. [[CrossRef](#)]
41. Andrieu, C.; Thoms, J. A Tutorial on Adaptive MCMC. *Stat. Comput.* **2008**, *18*, 343–373. [[CrossRef](#)]
42. Cheung, S.H.; Beck, J.L. Bayesian Model Updating Using Hybrid Monte Carlo Simulation with Application to Structural Dynamic Models with Many Uncertain Parameters. *J. Eng. Mech.* **2009**, *135*, 243–255. [[CrossRef](#)]
43. Geyer, C.J. *Markov Chain Monte Carlo Maximum Likelihood*; Interface Foundation of North America: Fairfax Station, VA, USA, 1991.
44. Neal, R.M. Sampling from Multimodal Distributions Using Tempered Transitions. *Stat. Comput.* **1996**, *6*, 353–366. [[CrossRef](#)]
45. Lam, H.F.; Yang, J.H.; Au, S.K. Markov Chain Monte Carlo-based Bayesian Method for Structural Model Updating and Damage Detection. *Struct. Control Health Monit.* **2018**, *25*, e2140. [[CrossRef](#)]
46. Zienkiewicz, O.C.; Zhu, J.Z. A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis. *Int. J. Numer. Methods Eng.* **1987**, *24*, 337–357. [[CrossRef](#)]

47. Prud'homme, C.; Rovas, D.V.; Veroy, K.; Machiels, L.; Maday, Y.; Patera, A.T.; Turinici, G. Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods. *J. Fluids Eng.* **2002**, *124*, 70–80. [[CrossRef](#)]
48. Ryckelynck, D.; Benzi, D.M. Multi-Level a Priori Hyper-Reduction of Mechanical Models Involving Internal Variables. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 1134–1142. [[CrossRef](#)]
49. Carlberg, K.; Farhat, C.; Cortial, J.; Amsallem, D. The GNAT Method for Nonlinear Model Reduction: Effective Implementation and Application to Computational Fluid Dynamics and Turbulent Flows. *J. Comput. Phys.* **2013**, *242*, 623–647. [[CrossRef](#)]
50. Kerfriden, P.; Ródenas, J.J.; Bordas, S.P.A. Certification of Projection-Based Reduced Order Modelling in Computational Homogenisation by the Constitutive Relation Error. *Int. J. Numer. Methods Eng.* **2014**, *97*, 395–422. [[CrossRef](#)]
51. Cui, T.; Marzouk, Y.M.; Willcox, K.E. Data-Driven Model Reduction for the Bayesian Solution of Inverse Problems. *Int. J. Numer. Methods Eng.* **2015**, *102*, 966–990. [[CrossRef](#)]
52. Hoang, K.; Kerfriden, P.; Bordas, S. A Fast, Certified and “Tuning Free” Two-Field Reduced Basis Method for the Metamodelling of Affinely-Parametrised Elasticity Problems. *Comput. Methods Appl. Mech. Eng.* **2016**, *298*, 121–158. [[CrossRef](#)]
53. Gelman, A.; Rubin, D.B. Inference from Iterative Simulation Using Multiple Sequences. *Stat. Sci.* **1992**, *7*, 457–472. [[CrossRef](#)]
54. Brooks, S.P.; Gelman, A. General Methods for Monitoring Convergence of Iterative Simulations. *J. Comput. Graph. Stat.* **1998**, *7*, 434–455.
55. Chamoin, L.; Florentin, E.; Pavot, S.; Visseq, V. Robust Goal-Oriented Error Estimation Based on the Constitutive Relation Error for Stochastic Problems. *Comput. Struct.* **2012**, *106*, 189–195. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).